January 1980

FMS–11/RSX
Release Notes

Order No. AA-H857A-TC

SUPERSESSION/UPDATE INFORMATION: This is a new manual.

OPERATING SYSTEM AND VERSION: RSX-11M V3.2
RSX-11M-PLUS V1

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation · maynard, massachusetts

First Printing, January 1980

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	PDT
DATATRIEVE	TRAX	

CONTENTS

1.0 INTRODUCTION

This document describes the installation procedures for FMS-11/RSX software, the FMS-11/RSX User Environment Test Package (UETP), information about using FMS-11 on a VT52 terminal, and information about the Form Driver Resident Library.

1.1 Using FMS-11/RSX

The following steps summarize how to use your FMS-11/RSX software:

1. Follow the installation procedures in Section 2 of this document.

2. Verify that your hardware and software are working properly by running the demonstration programs provided and by completing the UETP procedure described in Section 3 of this document.

3. Read the FMS-11/RSX Software Reference Manual to learn to use the FMS-11/RSX software components.

4. Use the extended examples that are referenced in the FMS-11/RSX Software Reference Manual to become acquainted with the software.

5. Write your own small FMS-11 application.

1.2 The FMS-11/RSX Documentation Set

The three documents for the FMS-11/RSX software and their order numbers are:

1. FMS-11/RSX Software Reference Manual AA-H855A-TC

2. FMS-11/RSX Mini-Reference AV-H856A-TC

3. FMS-11/RSX Release Notes AA-H857A-TC

2.0 INSTALLING FMS-11/RSX

The procedure for installing FMS-11 on an RSX-11M or an RSX-11M-PLUS system consists of copying the files from the distribution media to the device where they will be used and building the FMS-11 components. The entire process is controlled by an indirect command file.

When the command files finish executing, the FMS utilities can be copied to the system account and installed with the MCR INSTALL command.

You must be logged in under a privileged account in order for the installation to successfully complete. If you are installing FMS-11 on an RSX-11M-PLUS system, your terminal must be set for MCR mode commands. It is assumed that FMS-11 is being installed on the system device. All FMS-11 files are moved into account [30,10]. A VT100 is required to run the Form Editor and the demonstration programs and the UETP built as part of the installation procedure.

Approximately 3600 disk blocks are required to install FMS-11/RSX if all the demonstration programs are built. The demonstration program in a particular language is built only if the corresponding language processor is installed in the system.

In the installation procedure detailed below, the parameter 'ddn' is the name and unit number for the device FMS-11 is being installed on. The parameter 'dev' is the name and unit number of the device on which the distribution media is mounted.

2.1 Installation Procedure

```
MCR>ASN ddn:=SY:                    ! Make the installation device
                                    ! the default system device
MCR>UFD ddn:[30,10]                 ! Create the UFD for FMS-11
MCR>SET /UIC=[30,10]                ! Set the default UIC
```

For magnetic tapes on RSX-11M:

```
MCR>ALL dev:                        ! Allocate the drive
MCR>FLX SY:=dev:FMSINS.CMD/DO       ! Copy installation command file
```

For magnetic tapes on RSX-11M-PLUS:

```
MCR>MOU dev:/FOR                    ! Mount tape as foreign device
MCR>FLX SY:=dev:FMSINS.CMD/DO       ! Copy installation command file
```

For disks on both systems:

```
MCR>MOU dev:FMSRSX                  ! Mount the disk
MCR>PIP SY:/NV=dev:FMSINS.CMD       ! Copy installation command file
```

For all media:

```
MCR>@FMSINS                         ! Copy files from distribution
                                    ! media and build utilities
                                    ! and demo programs
```

The installation command file prompts for the distribution device. The FMS-11 files are copied to account [30,10] on the system device. The Form Editor (FED) and the Form Utility (FUT) are task built. The MACRO versions of the demonstration program and the UETP are built automatically. The COBOL, BASIC-PLUS-2, FORTRAN IV, and FORTRAN IV-PLUS versions of the demonstration program are built only if the corresponding compilers are installed in the system. The same is true for the FORTRAN IV and FORTRAN IV-PLUS versions of the UETP. If the FORTRAN IV-PLUS compiler is installed in the system, it is assumed that the file LB:[1,1]F4POTS.OLB exists as the FORTRAN IV-PLUS OTS. The installation command file asks the user if this is the case. If not, the FORTRAN IV-PLUS versions of the programs cannot be built. Otherwise, the command file queries the user as to whether FCS or RMS support is provided. The FORTRAN IV-PLUS demonstration program is then task built accordingly. The UETP program is only built if FCS support is present.

Two prebuilt Form Driver libraries are distributed with FMS-11: an object library using FCS (FDVLIB.OLB) and an object library using RMS (FDVLRM.OLB). The distributed versions of the Form Driver provide support for the VT100 only and require the full duplex terminal driver. Support for all features other than debug mode is included. If you wish to reconfigure the Form Driver, tailoring it to specific application requirements, execute the command file FDVBLD.CMD.

The system on which FED and FUT are run must include checkpointing to a system checkpoint file and the full duplex terminal driver.

2.2 Verifying Installation Procedures

The installation procedure is almost self-verifying. As part of the installation, demonstration programs are compiled and task built if the appropriate language processors are installed in the system. The programs MACDEM, FORDEM, F4PDEM, BASDEM and CBLDEM all implement the same application in MACRO, FORTRAN IV, FORTRAN IV-PLUS, BASIC-PLUS-2, and COBOL. Listings of these programs and a detailed description of the application are included in Appendix B of the FMS-11/RSX Software Reference Manual. To verify that these programs operate correctly, you should run them after the installation is complete. For example, to run the MACRO version of the demonstration program type the following:

```
MCR>SET /UIC=[30,10]
MCR>RUN MACDEM
```

Note that any time one of the demonstration programs distributed with FMS-11 is run (including the UETP), it is necessary that the default UIC be set to the account containing the form library and data files required by the program.

The demonstration programs implement a menu-driven application that allows you to enter customer information, parts descriptions, and employee information. The initial menu form allows the user to select one of the three functions or to exit. When the detail form(s) pertaining to a function have been completed, a second menu form is displayed. At this point the options are to repeat the same function, to return to the initial menu form, or to exit from the program. Each time a function is initiated, a file is created and the data collected is written sequentially into the file. The names of the files correspond to the functions: NEWCUS.DAT for customer information, PARTS.DAT for parts descriptions, and EMPLOY.DAT for employee information. You may inspect these files with an editor to verify the data.

To verify that the Form Utility (FUT) was built successfully, use it to get a directory of one of the form libraries distributed with FMS-11 and a listing of a form description. The following sample session with FUT gives a directory of the form library DEMLIB.FLB and a listing of the form PARTS contained in that library.

```
MCR>SET /UIC=[30,10]
MCR>RUN FUT
FUT>DEMLIB.FLB/LI

FUT V01.00
4-DEC-79

Library DL0:[30,10]DEMLIB.FLB;1 created: 4-DEC-79
Directory is 1 block long.
```

Form	Date	Impure area (bytes)
FIRST	4-DEC-79	369
CUSTPR	4-DEC-79	326
LAST	4-DEC-79	275
EMPLOY	4-DEC-79	812
PARTS	4-DEC-79	794
CUSTO	4-DEC-79	612

```
FUT>PARTS=DEMLIB.FLB/FD

DL0:[30,10]DEMLIB.FLB Form name? PARTS

DL0:[30,10]DEMLIB.FLB Form name? <CR>

FUT>
```

The file [30,10]PARTS.FMD contains the form description for the form PARTS. The file may be spooled to a line printer to obtain a hard copy listing.

To verify that the Form Editor (FED) was built correctly, use the example provided in Section 2.7 of the FMS-11/RSX Software Reference Manual to create a new form and modify an existing form.

3.0 USER ENVIRONMENT TEST PACKAGE (UETP)

The User Environment Test Package (UETP) for FMS-11/RSX is designed as an integrity test of the Form Management System. The intent is to verify that a typical FMS-11 application can be built and executed with the installed software. The UETP application is a Simple Inventory System, referred to throughout the remainder of this document as SIS.

The UETP is built as part of the installation procedure. The MACRO version (SIS.TSK) is built automatically. The FORTRAN IV (SISF.TSK) and FORTRAN IV-PLUS (SISF4P.TSK) versions are built only if the corresponding compilers are installed in the system. The UETP requires FORTRAN IV-PLUS with FCS support for file I/O. Therefore, a FORTRAN IV-PLUS version of the UETP is only built if the FORTRAN IV-PLUS OTS on the system (LB:[1,1]F4POTS.OLB) provides support for FCS rather than RMS.

The UETP is built to include a script processor for automatic execution. The script provided is designed to exercise a majority of the capabilities of the Form Driver component of FMS-11.

3.1 Running the UETP

To run the UETP, set the default UIC to [30,10]. Initialize the necessary application files by running the initialization program as follows:

```
MCR>RUN INITF
```

You can then run either the MACRO or one of the FORTRAN versions of the UETP. To run the FORTRAN IV version, for example, type

```
MCR RUN SISF
```

Since the program executes under script control, character input is taken from the script file rather than the terminal keyboard. Script characters are supplied every half second to simulate human typing. Some delays have been included in the script to facilitate reading. The progress of the script can be controlled from the keyboard. Type 'S' to temporarily halt the script; type 'G' to resume the script. After halting the script by typing 'S', you can step through it one character at a time by typing <SPACE>. Since escape sequences are several characters long, it is necessary in some cases to step through more than one character before any action is taken by the Form Driver. The script can be aborted at any time by typing 'Z' (not control Z). After aborting the script, all subsequent input is taken from the keyboard rather than the file. Should you abort in the middle of an escape sequence, the Form Driver may signal an error (by sounding the bell) in response to the first few characters entered manually, but it should not persist.

3.2 UETP Step-by-Step Procedure

The following description of the execution of the script is divided so that each step corresponds to a single form displayed by the application. Within each step, the input from the script and the resulting actions are described.

STEP 1 - SIS INTRODUCTION

1. The script types the <HELP> key (the "PF2" key on the keypad) and a one line explanation of how to respond to the prompt NEXT? is displayed on the last line of the screen.

2. The script types <HELP> again, and FDV displays the HELP form that is associated with the introduction form.

STEP 2 - INTRODUCTION HELP FORM

The script types the <RETURN> key, and FDV displays the introduction form again. The cursor is at the NEXT? prompt.

STEP 3 - SIS INTRODUCTION

The script accepts the default response (4) to request a correction form. SIS processes the response, and the correction form is displayed.

STEP 4 - CORRECTION REQUEST

The script overrides the default response (N) and types D to request the form for changing inventory descriptions. SIS processes the response and displays the requested form.

STEP 5 - CHANGE DESCRIPTION

1. The script types a stock number, advances to the next input field ("Description") by typing the <TAB> key, and types the name of the item.

2. The script types in several new descriptions, advancing from field to field by using the <TAB> key, and types <RETURN> when the form is complete.

3. SIS processes the entries and displays the next form, a menu of SIS processes.

STEP 6 - MENU

The script overrides the original default response (0) and types 4 to request the correction form again.

STEP 7 - CORRECTION REQUEST

The script types I to request the form for changing inventory quantities, and the form is displayed.

STEP 8 - CHANGE INVENTORY QUANTITY

1. The script types a stock number and the <TAB> key. SIS displays the corresponding description.

2. SIS advances the cursor to the "Quantity in stock" field.

3. The script types a quantity and the <TAB> key. The initial quantity for each item is 100.

4. SIS advances the cursor to the next line of the form.

5. The script types several stock numbers and quantities and then types <RETURN>. SIS processes the entries, and the menu form is displayed.

STEP 9 - MENU

SIS has been designed to modify the default response to the menu form. The original default response is 0. After any other response is typed, SIS changes the default response to the last response typed. Therefore, the default is 4 at this point because the script typed 4 in Step 6.

The script accepts the default by typing only the <RETURN> key, and the correction form is displayed.

STEP 10 - CORRECTION REQUEST

The script types A to request the form for changing account information, and the form is displayed.

STEP 11 - CHANGE ACCOUNT

1. The script completes each field and types the <TAB> key to advance to the next field.

2. When the information for the first account is complete, the script accepts the default response (Y) for the prompt MORE ACCOUNTS? by typing <RETURN>.

3. SIS processes the new account information, displays a blank version of the same form, and places the cursor at the first field in the form.

4. When the information for the second account is complete, the script overrides the default response to the prompt MORE ACCOUNTS? by typing N and <RETURN>.

5. SIS processes the new account information, and the menu form is displayed.

STEP 12 - MENU

The script types 3 to use the inventory status request form and the form is displayed.

STEP 13 - STATUS

1. The script enters several stock numbers and types <TAB> after each one.

2. SIS displays the corresponding descriptions and quantities in stock. At this point in the script, there are 100 pieces in stock for each item.

3. After checking the inventory status of each item, the script types <RETURN> to indicate that work with the form has been finished.

4. The menu form is displayed.

The "Stock Number," "Description," and "Quantity in Stock" fields in the inventory status request form are indexed fields. Each data line in the form contains the same fields. Within a line, SIS references the fields by name. Within the form as a whole, SIS references lines by indexes, with the first data line having the index 1, the second line the index 2, and so on.

STEP 14 - MENU

The script types 1 to use the form for recording incoming shipments, and the form is displayed.

STEP 15 - RECEIVING FORM

1. The script types the supplier's account number and <TAB>. SIS displays account information from the account file and locates the cursor at the invoice number field.

2. The script types an invoice number and <TAB>. SIS locates the cursor at the first "STOCK #" field.

3. For each item received, the script types the stock number and <TAB>.

4. SIS displays the corresponding description and places the cursor in the "QUANTITY" field.

5. The script types the quantity received and <TAB>. SIS places the cursor at the "STOCK #" field in the next line of the form.

6. On the fourth line, the script types the stock number 500 by mistake. However, the typing error is not detected until after <TAB> is typed.

7. With the cursor at the "QUANTITY" field, the script types the <BACKSPACE> key to move the cursor back to the "STOCK #" field.

8. The script uses the <DELETE> key to erase the incorrect stock number.

9. When the script types the correct stock number (501) and <TAB>, SIS corrects the description and relocates the cursor at the "QUANTITY" field.

10. After recording all items received, the script types <RETURN>. SIS processes the entries, updates the data base and the menu form is displayed.

STEP 16 - MENU

The script types 3 to request the inventory status request form.

STEP 17 - STATUS

1. The script checks the new inventory status of several items by typing their stock numbers and the <TAB> key.

2. After each stock number is complete, SIS displays the description of the item and the current inventory quantity. The quantities reflect the incoming shipments recorded in Step 15.

3. The script types <RETURN> to indicate that work with the status form is complete, and the menu form is displayed.

STEP 18 - MENU

The script types 2 to request the form for outgoing shipments, and the form is displayed.

STEP 19 - SHIPPING FORM

1. The script types the customer's account number and <TAB>. SIS displays account information from the account file and locates the cursor at the "PICKING LIST #" field.

2. The script types a picking list number and <TAB>. SIS locates the cursor at the first stock number field.

3. For each of several items in the shipment, the script types a stock number and <TAB>. SIS supplies the description and locates the cursor at the "QUANTITY" field.

4. The script types the quantity shipped and <TAB>, and SIS locates the cursor at the next "STOCK #" field.

5. After recording all of the items in the shipment, the script types <RETURN>. SIS processes the entries, updates the data base, and the menu form is displayed.

STEP 20 - MENU

The script types 3 to request the inventory status request form.

STEP 21 - STATUS

The script requests the inventory status for several items. The quantities reflect the outgoing shipments recorded in Step 19. After the script types <RETURN>, the menu form is displayed.

STEP 22 - MENU

The script types 4 to request the correction form and the form is displayed.

STEP 23 - CORRECTION REQUEST

The script types D to request the form for changing inventory descriptions and the form is displayed.

STEP 24 - CHANGE DESCRIPTION

To prepare for an incoming shipment of a new inventory item, the script types the item's new stock number and description. The script then types <RETURN>, and the menu form is displayed.

STEP 25 - MENU

The script types 1 to use the form for recording incoming shipments and the form is displayed.

STEP 26 - RECEIVING FORM

The script types the stock number and quantity of the incoming shipment. SIS displays the description of the item. The script types <RETURN>, and the menu form is displayed.

STEP 27 - MENU

The script types 3 to request the inventory status request form, and the form is displayed.

STEP 28 - STATUS

The script types the stock numbers for several items in the inventory, including the new item recorded in Steps 24 and 26. SIS displays descriptions and current inventory quantities. The script types <RETURN>, and the menu form is displayed.

STEP 29 - MENU

The script types 1 to use the form for recording incoming shipments, and the form is displayed.

STEP 30 - RECEIVING FORM

1. The script types the account number 1 by mistake. However, the error is not detected until after <TAB> is typed and SIS displays the information for account number 1.

2. With the cursor at the "INVOICE NUMBER" field, the script types the <BACKSPACE> key to move the cursor back to the "ACCOUNT NUMBER" field.

3. The script uses the <DELETE> key to erase the account number.

4. The script types the correct account number (500), and SIS displays the account information for that account. SIS has not processed the incorrect data because the error was corrected before typing the <RETURN> key. Therefore, trace records for this transaction will show only the correct account number (500).

5. The script types an invoice number and <TAB>.

6. To demonstrate how a scrolling area within a form is displayed, the script fills all of the displayed blank lines in the form.

7. When the script types <TAB> after the quantity received in the last line, the entire field of stock information scrolls upward. The top line scrolls out of sight, and a new blank line scrolls into view at the bottom.

8. The script continues to demonstrate scrolling by typing <BACKSPACE> to move the cursor toward the top line of stock information. Lines that were not displayed scroll downward into view, and the bottom lines scroll out of sight.

   This step also demonstrates a special programming technique. SIS has been designed to process <BACKSPACE> in this context by locating the cursor at the preceding "STOCK #" field in all cases. Therefore, the cursor is prevented from moving backward from a "STOCK #" field to the preceding "QUANTITY" field.

9. When the demonstration of scrolling is finished, the script types <RETURN>, and the menu form is displayed.

STEP 31 - MENU

The last step for the automatic script is to type 6 and <RETURN>. The menu form does not show that 6 is a valid choice, but SIS can nevertheless process the response. This technique may be useful in another application that is designed with special features for use by a site supervisor. The supervisor can use the features without requiring site operators to learn about them.

SIS displays a special menu form that signals the end of the automatic script. At this point you can experiment with the UETP application by requesting the forms you want and by typing your own information in their fields.

To stop the UETP application, type <RETURN> until the menu form is displayed and then type 5 and <RETURN>. When the UETP application stops, the monitor prompt is displayed.

4.0 VT52 Support

FMS-11 was designed to take advantage of the features of the VT100 terminal. The Form Editor requires a VT100 terminal; it will not operate on a VT52. The Form Driver, however, can be built to support either the VT100 or the VT52. Support for both types of terminals cannot exist in a single Form Driver library; separate libraries are required. The Form Driver libraries distributed with FMS-11 kit (FDVLIB.OLB for FCS and FDVLRM.OLB for RMS) provide VT100 support. If you wish to create a Form Driver with VT52 support, you mus go through the Form Driver configuration procedure, invoked by executing the command file FDVBLD.CMD. The user is prompted as to which Form Driver options are desired, including VT52 support. If VT52 support is selected, the Form Driver libraries F52LIB.OLB for FCS and F52LRM.OLB for RMS are created. They should be included in the task build for form applications exactly as the VT100 versions of the libraries are.

The VT52 version of the Form Driver provides essentially the same capabilities as the VT100 version. The following are the differences:

1. VT100 video attributes

   The VT52 does not provide support for reverse, bold, blinking, underscore, 132 column mode, or reverse screen background. The VT52 Form Driver ignores these attributes when specified for a form.

2. Scrolling

   The VT52 does not support split screen scrolling. However, the VT52 Form Driver will emulate scrolling in software by redisplaying all fields in a scrolled area when the area is scrolled.

3. Function keys

   Due to the differences in the alternate keypads, the positions of the following three function keys are different for the VT100 and the VT52 as specified below:

	VT100	VT52
Insert/Overstrike	PF1	CTRL/A
Exit scrolled area backward	PF3	Blue key
Exit scrolled area forward	PF4	Gray key

   The HELP key is in the same position on both terminals: the PF2 key on the VT100 and the Red key on the VT52. All other functions are the same for the VT52 as the VT100.

After creating Form Driver libraries with VT52 support, you may build the FMS-11 demonstration programs to run on the VT52 by task building with the correct library. However, the script provided for the UETP is for the VT100 version of the Form Driver and will not work with the VT52 version.

5.0 FORM DRIVER RESIDENT LIBRARY

The Form Driver with FCS support may be built as a resident library. However, the version with RMS support cannot be built as a resident library.

Task build command files are distributed to build both the VT100 and VT52 versions of the Form Driver as resident libraries (FDVRES.CMD and F52RES.CMD). Before a Form Driver resident library can be built, the Form Driver configuration procedure must be executed to create the files required for the resident library.

Please cut along this line.

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Did you find errors in this manual? If so, specify the error and the page number.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Please indicate the type of reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify)_____

Name_____ Date_____

Organization_____

Street_____

City_____ State_____ Zip Code or Country_____

Do Not Tear - Fold Here and Tape

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

RT/C SOFTWARE PUBLICATIONS ML 5-5/E45
DIGITAL EQUIPMENT CORPORATION
146 MAIN STREET
MAYNARD, MASSACHUSETTS 01754

Do Not Tear - Fold Here

Cut Along Dotted Line